

Nattan MAISONNEUVE–FLEURY
BTS SIO 1
Groupe B

10/11/2025

Compte Rendu - Bataille Navale

Sommaire :

Introduction :	1
Etape 1 :	1
Etape 2 :	1
Etape 3 :	3
Etape 4 :	3
Etape 5 :	4
Etape 6 :	5
Etape 7 :	6
Etape 8 :	6
Conclusion :	7

Introduction :

Dans ce TP, nous allons voir comment transformer plusieurs algorithmes en Java afin de nous familiariser avec ce langage.

Etape 1 :

Dans cette première étape, nous allons déclarer deux tableaux de 5 lignes par 5 colonnes sous forme de variable afin de pouvoir les réutiliser facilement par la suite. Voici le code :

```
int tabjoueur[][] = new int [5][5];  
int tabordi[][] = new int [5][5];
```

Ensuite on veut initialiser toutes les cases à zéro. Pour ce faire, on va parcourir chaque ligne et dans chacune d'elle, on va parcourir le nombre total de colonnes. Voici le code :

```
int i = 0;  
while (i < tabordi.length) { // tant que i est inférieur à longueur  
    tabordi  
    int j = 0; //initialisation de la variable j a 0  
    while (j < tabordi[i].length) { // tant que j est inférieur à  
        nombre de colonne  
        tabordi[i][j] = 0; // initialisation de la variable a 0  
        j++; //j+1  
    }  
    i++; //i+1  
}
```

Etape 2 :

Dans cette étape il faut tout d'abord déclarer une grande boucle qui dit "tant que les 5 pions n'ont pas été placés alors on demande à l'utilisateur de rentrer une ligne comprise entre 0 et 4 et une colonne entre 0 et 4 ce qui nous donneras une case. (On prend l'intervalle 0 et 4 car le tableau a pour premier indice 0. Si nous commençons à 1, il aurait fallu faire -1 a chaque fois que l'utilisateur rentre un chiffre.) Ensuite, on vérifie si les chiffres entrés sont bien compris entre 0 et 4. Si ce n'est pas le cas, on demande à l'utilisateur de re-rentre des

chiffres. Si les chiffres sont compris, alors on vérifie que l'utilisateur n'a pas déjà placé un pion sur cette case. Voici le code :

```
do{ //faire
    while (i < tabordi.length) { // tant que i est inférieur à
longueur tabordi
        int j = 0; //initialisation de la variable j a 0
            while (j < tabordi[i].length) { // tant que j est
inférieur à nombre de colonne
                tabordi[i][j] = 0; // initialisation de la variable
a 0
                j++; //j+1
            }
            i++; //i+1
        }

    while (
        choixJoueurLigne < 0 || choixJoueurLigne > 4 ||
        choixJoueurColonne < 0 || choixJoueurColonne > 4 ||
        tabjoueur[choixJoueurLigne][choixJoueurColonne] != 0
    ) { // tant que....
        System.out.println("Sur quelle ligne voulez-vous placer
votre pion ? (0 à 4)"); //afficher la phrase
        choixJoueurLigne = sc.nextInt();//enregistre l'information
de l'utilisateur dans la variable...

        System.out.println("Sur quelle colonne voulez-vous placer
votre pion ? (0 à 4)");//afficher la phrase
        choixJoueurColonne = sc.nextInt();//enregistre
l'information de l'utilisateur dans la variable...

        if (choixJoueurLigne < 0 || choixJoueurLigne > 4 ||
choixJoueurColonne < 0 || choixJoueurColonne > 4) { //si...
            System.out.println("Valeurs non comprises entre 0 et
4");//afficher la phrase
        } else if (tabjoueur[choixJoueurLigne][choixJoueurColonne]
!= 0) { //sinon si...
            System.out.println("Case déjà prise.");//afficher la
phrase
        }
    }

    tabjoueur[choixJoueurLigne][choixJoueurColonne] = 1;//la
variable... est initialiser a 1
```

```

System.out.println("Le pion a bien été placé");//afficher la phrase
    compteur++; //compteur +1
} while (compteur != 5); //tant que compteur différent de 5

```

Il faut bien sûr déclarer toutes les variables :

```

int choixJoueurLigne = 0;
int choixJoueurColonne = 0;
int compteur = 0;

```

Etape 3 :

Dans cette étape on fait choisir aléatoirement deux chiffres à l'ordinateur, pour ce faire, on utilise une fonction Math.random Voici le code :

```

do { // faire
    do { //faire
        choixOrdiLigne = (int) (Math.random() * 5); // ligne aléatoire
        entre 0 et 4
        choixOrdiColonne = (int) (Math.random() * 5); // colonne aléatoire
        entre 0 et 4
    } while (tabordi[choixOrdiLigne][choixOrdiColonne] != 0); //tant que
    la case n'est pas à 1

    tabordi[choixOrdiLigne][choixOrdiColonne] = 1; //la case prend por
    valeur 1

    compteur++; // compteur +1
} while (compteur != 5); // tant que compteur différent de 5

```

Etape 4 :

On peut créer une procédure qui nous permettra d'afficher le tableau avec nos "bateau" pour que l'utilisateur voit bien où sont placés ses pions. Voici le code :

```

static void affichagetab(int[][] tab, int nbcase) {
    System.out.print(" "); //espace pour que les lignes soient
    bien alignées

```

```

        for (int col = 0; col < nbcase; col++) { //pour la variable col
            tant que col est inférieur a nbcase. col + 1 a chaque tour
            System.out.print(col + " "); //afficher un espace
        }
        System.out.println(); //sert a passer a ligne suivante

        for (int i = 0; i < nbcase; i++) { //pour i tant que i
            inférieur à nbcase. i + 1 a chaque tour
            System.out.print(i + " "); // Numéro de ligne
            for (int j = 0; j < nbcase; j++) { //pour j tant que j
                inférieur à nbcase. j + 1 a chaque tour
                if (tab[i][j] == 1) { //la variable s'initialise a 1
                    System.out.print("O "); // pion du joueur
                } else { //sinon
                    System.out.print("~ "); // case vide
                }
            }
            System.out.println();
        }
    }
}

```

```

Tableau du joueur :
  0 1 2 3 4
0 ~ ~ ~ ~ ~
1 0 ~ ~ 0 ~
2 ~ ~ ~ 0 ~
3 ~ ~ ~ 0 ~
4 ~ 0 ~ ~ ~

```

Etape 5 :

```

static void affichageCache(int[][] tab, int nbcase) {
    System.out.print(" "); //espace pour que les lignes soient bien
    alignées
    for (int indiceColonne = 0; indiceColonne < nbcase;
        indiceColonne++) { //pour la variable indiceColonne tant que
        indiceColonne est inférieur a nbcase. indiceColonne + 1 a chaque
        tour
        System.out.print(indiceColonne + " "); //afficher un espace après la
        variable
    }

    System.out.println(); //sert a passer a ligne suivante
}

```

```

for (int i = 0; i < nbcase; i++) {
    System.out.print(i + " "); // numéro de ligne
    for (int j = 0; j < nbcase; j++) { //pour j tant que j
        inférieur à nbcase. j + 1 a chaque tour
        if (tab[i][j] == 0) { //si la variable est égale à 0
            System.out.print("? "); // case non découverte
        } else if (tab[i][j] == 2) { //sinon si égale a 2
            System.out.print("O "); // case avec bateau
        } else if (tab[i][j] == 3) { //sinon si égale a 3
            System.out.print("X "); // case vide
        }
    }
    System.out.println(); //sert a passer a ligne suivante
}

```

Etape 6 :

```

// ETAPE 6 - Tour du joueur
System.out.println("\nÀ vous de jouer !"); // Tour du joueur
affichageCache(tabordi, nbcase: 5); // Affichage du tableau de l'ordi caché

int tirLigne, tirColonne; // Coordonnées du tir du joueur
do { // Demande des coordonnées tant que les valeurs ne sont pas correctes
    System.out.println("Sur quelle ligne voulez-vous tirer ? (0 à 4)"); // Demande des coordonnées
    tirLigne = sc.nextInt(); // Lecture de la ligne

    System.out.println("Sur quelle colonne voulez-vous tirer ? (0 à 4)"); // Demande des coordonnées
    tirColonne = sc.nextInt(); // Lecture de la colonne

    if (tirLigne < 0 || tirLigne > 4 || tirColonne < 0 || tirColonne > 4) { // Vérification des coordonnées
        System.out.println("Valeurs non comprises entre 0 et 4 !"); // Message d'erreur
    }
} while (tirLigne < 0 || tirLigne > 4 || tirColonne < 0 || tirColonne > 4); // Vérification des coordonnées

System.out.println("Tir en cours..."); // Message de tir
try { // Pause de 2 secondes pour simuler le tir
    Thread.sleep(2000); // Pause de 2 secondes
} catch (InterruptedException e) { // Gestion de l'exception
    e.printStackTrace(); // Affichage de l'exception
}

if (tabordi[tirLigne][tirColonne] == 0) { // Si le tir est un échec
    System.out.println("Raté !"); // Message de raté
    tabordi[tirLigne][tirColonne] = 3; // Marquage de la case comme ratée
} else if (tabordi[tirLigne][tirColonne] == 1) { // Si le tir est un succès
    System.out.println("Touché !"); // Message de touché
    tabordi[tirLigne][tirColonne] = 2; // Marquage de la case comme touchée
    nbPionTrouveJoueur++; // Incrémentation du compteur de pions trouvés
} else { // Si le tir est sur une case déjà découverte
    System.out.println("Tir à blanc (vous avez déjà tiré ici !"); // Message de tir à blanc
}

System.out.println("\nTableau de l'ordinateur :"); // Affichage du tableau de l'ordi mis à jour
affichageCache(tabordi, nbcase: 5); // Affichage du tableau de l'ordi caché mis à jour

if (nbPionTrouveJoueur == 5) { // Si le joueur a trouvé tous les pions
    System.out.println("Bravo, vous avez trouvé tous les pions !"); // Message de victoire
    finPartie = true; // Fin de la partie
    break; // Sortie de la boucle
}

```

Etape 7 :

```
System.out.println("\nTour de l'ordinateur..."); // Tour de l'ordinateur
System.out.println("L'ordinateur réfléchit..."); // Message de réflexion

int tirOrdiLigne, tirOrdiColonne; // Coordonnées du tir de l'ordinateur
do { // Génération des coordonnées tant que les valeurs ne sont pas correctes
    tirOrdiLigne = (int)(Math.random() * 5); // Génération aléatoire de la ligne
    tirOrdiColonne = (int)(Math.random() * 5); // Génération aléatoire de la colonne
} while (tabJoueur[tirOrdiLigne][tirOrdiColonne] != 0 && tabJoueur[tirOrdiLigne][tirOrdiColonne] != 1); // Vérification des coordonnées

System.out.println("Tir en cours..."); // Message de tir
try { // Pause de 2 secondes pour simuler le tir
    Thread.sleep(2000); // Pause de 2 secondes
} catch (InterruptedException e) { // Gestion de l'exception
    e.printStackTrace(); // Affichage de l'exception
}

if (tabJoueur[tirOrdiLigne][tirOrdiColonne] == 0) { // Si le tir est un échec
    System.out.println("L'ordinateur a tiré en (" + tirOrdiLigne + ", " + tirOrdiColonne + ") → Raté !"); // Message de raté
    tabJoueur[tirOrdiLigne][tirOrdiColonne] = 3; // Marquage de la case comme ratée
} else if (tabJoueur[tirOrdiLigne][tirOrdiColonne] == 1) { // Si le tir est un succès
    System.out.println("L'ordinateur a tiré en (" + tirOrdiLigne + ", " + tirOrdiColonne + ") → Touché !"); // Message de touché
    tabJoueur[tirOrdiLigne][tirOrdiColonne] = 2; // Marquage de la case comme touchée
    nbPionTrouveOrdi++; // Incrémentation du compteur de pions trouvés
} else { // Si le tir est sur une case déjà découverte
    System.out.println("L'ordinateur a tiré sur une case déjà découverte !"); // Message de tir à blanc
}

System.out.println("\nTableau du joueur mis à jour :"); // Affichage du tableau du joueur mis à jour
afficheTab(tabJoueur, nbCase: 5); // Affichage du tableau du joueur mis à jour

if (nbPionTrouveOrdi == 5) { // Si l'ordinateur a trouvé tous les pions
    System.out.println("L'ordinateur a trouvé tous vos pions... Vous avez perdu !"); // Message de défaite
    finPartieOrdi = true; // Fin de la partie
    break; // Sortie de la boucle
}
```

Etape 8 :

```
        System.out.println("\n=====");
if (finPartie) {
    System.out.println("Félicitations, vous avez gagné la partie");
} else if (finPartieOrdi) {
    System.out.println("L'ordinateur a gagné la partie");
}

    System.out.println("Voulez-vous rejouer ? (o/n)");
    char rejouer = sc.next().toLowerCase().charAt(0);
    if (rejouer != 'o') {
        recommencer = false;
        System.out.println("FIN DU JEU");
    }
}
```

Pour pouvoir rejouer il faut entourer TOUT le code avec : `while (recommencer) {`

En déclarant une variable : `boolean recommencer = true;`

Conclusion :

J'ai trouvé ce TP facile au début car il y avait peu de lignes donc il était facile de s'y retrouver, mais la fin du tp a été pour ma part compliqué, car il y avait beaucoup de lignes et si on se perdait une fois on était perdu pour toujours... Malgré tout, ce TP a permis d'apprendre de nouvelles choses (boolean en variable, bien séparé le code, les résultats...)